

REDUNDANCIA FENOTIPICA Y OPERADORES GENETICOS NO CONVENCIONALES PARA MOVERSE DE OPTIMOS LOCALES A OPTIMOS GLOBALES EN AMBIENTES ESTATICOS

Aragón Victoria , Esquivel S. C.,

Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)
Facultad de Ciencias Físico, Matemáticas y Naturales
Departamento de Informática
Universidad Nacional de San Luis
Ejercito de los Andes 950 – Local 106
{vsaragon, esquivel}@unsl.edu.ar

Resumen

El trabajo presenta un modelo evolutivo para ambientes estáticos con el objetivo de estudiar el comportamiento de ciertos operadores sujetos a una representación y recombinación especial para permitir el movimiento, de una población en evolución, desde un óptimo local hacia uno global. El modelo trabaja a nivel fenotípico y los fenos son números reales. El cromosoma de cada individuo está compuesto por fenos activos (influyen en el fitness) y fenos redundantes (no influyen en el fitness). El rol de los fenos varía durante la evolución a causa de la aplicación de operadores específicos. El modelo implementa los operadores genéticos básicos: recombinación y mutación. Pero también implementa otros operadores adicionales: crisis, recrudescencia, transposición y transición [3]. Resultados preliminares de las simulaciones del modelo se describen y discuten.

PALABRAS CLAVES: Computación evolutiva, redundancia fenotípica, operadores evolutivos.

1 INTRODUCCION

Una de las razones principales para imitar, en los sistemas artificiales, a la evolución natural es su capacidad intrínseca de adaptación. Esta característica de la evolución puede ser utilizada en muchos problemas de optimización del mundo real siempre que el objetivo de la optimización se exprese como mejoras graduales de las soluciones y no como convergencia “al mejor de todos”.

Como en todo modelo, también en los enfoques evolutivos, existe una simplificación de la realidad que se trata de imitar a través de la simulación. Es conocido que sólo una pequeña parte de los *genotipos* de los organismos vivos codifican las características expresables, *fenotipos*, de dichos organismos [1, 7]. A tales genes es a los que, en el modelo que se presenta, se los denomina *genes activos*, el resto se consideran como *redundantes o latentes*. Esta partición suele no ser fija, en el sentido que como efecto de algún proceso de reorganización de los cromosomas, algún gene redundante puede transformarse en activo y, a partir de dicho momento, influenciar el fenotipo de un individuo. Esta característica de la evolución natural casi no ha sido estudiada por la comunidad de computación evolutiva [3].

Otro concepto que incluye el modelo es el de macromutación, es decir el proceso que produce modificaciones radicales en el espacio fenotípico. Los biólogos suelen coincidir en que el fenómeno de la macromutación está presente en la naturaleza [5, 1].

Uno de los problemas que se puede presentar durante el proceso de optimización, usando heurísticas evolutivas, es que los algoritmos converjan a un óptimo local cuando el fitness landscape posee muchos de estos óptimos.

Tomando como base los resultados de los trabajos realizados por Halina Kwasnicka [2, 3, 4] el presente trabajo modifica el modelo presentado por dicha investigadora con respecto a la representación, el operador de recombinación y la complejidad de las funciones tratadas con la finalidad de estudiar la capacidad del modelo evolutivo propuesto para moverse de óptimos locales hacia el óptimo global durante el proceso de evolución de una población.

El resto del trabajo está organizado de la siguiente manera: En la sección 2 se describe el modelo propuesto en cuanto a representación y definición de operadores, en la sección 3 se describe brevemente el conjunto de funciones de test utilizadas, en la sección 4 se detallan las características de los experimentos, la sección 5 presenta los resultados obtenidos y, finalmente, en la sección 6 se presentan las conclusiones y líneas de investigación corriente y futura.

2. Modelo Evolutivo

2.1 Representación

La población está compuesta por un número constante, N , de individuos. Todos los individuos constan de un único cromosoma, el cual es directamente el vector de fenos, es decir, en el modelo no se considera el nivel genotípico sino sólo el fenotípico. El cromosoma de un individuo está compuesto por fenos activos y fenos redundantes (latentes). Cada fen en el cromosoma es un valor real que pertenece al intervalo $[-1.0, 1.0]$. Los fenos activos representan las coordenadas de la función a optimizar. Cada cromosoma posee n fenos activos, donde n es la cantidad de argumentos de la función y $n*10$ fenos redundantes. Los fenos activos son los únicos que tienen influencia en el fenotipo de un individuo. El número de fenos redundantes, en cada individuo, se mantiene constante durante toda la evolución. La propiedad de un fen, en cuanto a ser activo o redundante, la define la posición que éste ocupa dentro del cromosoma. Un individuo está representado por:

$$I = \langle f_1, f_2, \dots, f_n, fr_{1, n+1}, \dots, fr_{n, n+n*10} \rangle$$

Por ejemplo, si la función a optimizar tiene dos argumentos de entrada entonces la cantidad de fenes activos es dos (cada uno representa uno de los argumentos de entrada), la cantidad de fenes redundantes es 20 (10 fenes redundantes para cada fen activo). En cambio si la función tiene 10 argumentos de entrada entonces la cantidad de fenes activos es 10 mientras que la cantidad de fenes redundantes es 100.

Supuesta $f(x,y)$ un cromosoma para un individuo representando una solución puede verse gráficamente:

fenes activos		fenes redundantes asociados al f_1			fenes redundantes asociados al f_2		
f_1	f_2	$fr_{1,1}$...	$fr_{1,10}$	$fr_{2,1}$...	$fr_{2,10}$

2.2 Operadores

Selección: Sólo se aplica selección para la reproducción de los individuos. El modelo implementa la selección proporcional.

Recombinación: La recombinación opera sobre los cromosomas. El operador de crossover que se usa se desarrolló especialmente para la representación de los individuos definida en el punto anterior. Dos individuos seleccionados, a través del operador de selección, intercambian parte de sus fenes con una probabilidad p_{recomb} . La recombinación entre los individuos se lleva a cabo en la forma descrita en el siguiente pseudo-código:

```

function recombinacion (in cromosoma padre1, padre2; in int k; out cromosoma hijo1, hijo2)
{
    /* k ∈ [1, n] donde n = cantidad de fenes activos de un individuo */
    /* recombinación de fenes activos */
    para i de 1 a k hacer
        copiar en la posición i del hijo1 el valor del fen activo que está en la posición i
        del padre1
        copiar en la posición i del hijo2 el valor del fen activo que está en la posición i
        del padre2
    finpara
    para i de k+1 a n hacer
        copiar en la posición i del hijo1 el valor del fen activo que está en la posición i
        del padre2
        copiar en la posición i del hijo2 el valor del fen activo que está en la posición i
        del padre1
    finpara
    /* recombinación de fenes redundantes */
}

```

Cada vez que se copia un fen activo de un padre a un hijo, también se copian en el hijo todos los fenes redundantes correspondientes a dicho fen activo del padre en la misma posición que éstos ocupaban en el cromosoma padre.

}

Los hijos pasan a formar parte de la nueva población mientras que los padres sólo pasarán a formar parte ella si no hubo recombinación entre ambos.

Ejemplo de la aplicación del crossover de los padres padre¹ y padre², generando hijo¹ e hijo². Se asume que la función posee dos argumentos de entrada, con lo cual se tienen 2 fenes activos y 20 fenes redundantes y $k = 1$.

padre¹

f_1^1	f_2^1	$fr_{1,1}^1$...	$fr_{1,10}^1$	$fr_{2,1}^1$...	$fr_{2,10}^1$
---------	---------	--------------	-----	---------------	--------------	-----	---------------

padre²

f_1^2	f_2^2	$fr_{1,1}^2$...	$fr_{1,10}^2$	$fr_{2,1}^2$...	$fr_{2,10}^2$
---------	---------	--------------	-----	---------------	--------------	-----	---------------

hijo¹

f_1^1	f_2^2	$fr_{1,1}^1$...	$fr_{1,10}^1$	$fr_{2,1}^2$...	$fr_{2,10}^2$
---------	---------	--------------	-----	---------------	--------------	-----	---------------

hijo²

f_1^2	f_2^1	$fr_{1,1}^2$...	$fr_{1,10}^2$	$fr_{2,1}^1$...	$fr_{2,10}^1$
---------	---------	--------------	-----	---------------	--------------	-----	---------------

Mutación: El operador de mutación opera sobre los fenes de un cromosoma, sean activos o redundantes, este operador modifica aleatoriamente el valor de un fen. Cada fen muta en forma independiente con una probabilidad p_{mut} . Durante el proceso de mutación el valor del fen se reemplaza por otro valor escogido, en forma aleatoria, dentro del rango de valores permitidos para dicho fen.

Transposición: El operador de transposición actúa sobre los fenes. Aquí, uno de los fenes redundantes asociado con un fen activo se vuelve activo y el activo se transforma en redundante con una probabilidad p_{transp} , con lo cual el fen que antes no tenía influencia sobre el fenotipo pasa a ser significativo sobre dicho espacio.

Por ejemplo, al fen redundante $fr_{1,3}^i$ del individuo i se le debe aplicar el operador de transposición (continuando con el ejemplo de una función de dos variables), el individuo i se transforma en el individuo i' :

I^i

f_1^i	f_2^i	$fr_{1,1}^i$...	$fr_{1,3}^i$...	$fr_{1,10}^i$	$fr_{2,1}^i$...	$fr_{2,10}^i$
---------	---------	--------------	-----	--------------	-----	---------------	--------------	-----	---------------

$I^{i'}$

$fr_{1,3}^{i'}$	$f_1^{i'}$	$f_2^{i'}$	$fr_{1,1}^{i'}$...	$f_1^{i'}$	$fr_{1,3}^{i'}$...	$fr_{1,10}^{i'}$	$fr_{2,1}^{i'}$...	$fr_{2,10}^{i'}$
-----------------	------------	------------	-----------------	-----	------------	-----------------	-----	------------------	-----------------	-----	------------------

Transición: El operador de transición actúa sobre los fenos, causa que un fen activo, seleccionado aleatoriamente, de un individuo sea copiado como fen redundante asociado con el correspondiente fen activo de un segundo individuo también seleccionado aleatoriamente. Este proceso provoca que se pierda el fen sobrescrito en esa posición. El operador se aplica con una probabilidad p_{trans} .

Continuado con el caso de una función de dos variables sea el fen activo 1 del individuo i al que se le aplica el operador de transposición. Tal fen se copiará en el individuo j , en cualquiera de las posiciones de fenos redundantes asociados con el fen 1 siempre y cuando esté comprendida entre el $fr_{1,1}$ y $fr_{1,10}$. Sea también que la posición redundante seleccionada fue $fr_{1,3}$.

I^i

f_1^i	f_2^i	$fr_{1,1}^i$...	$fr_{1,10}^i$	$fr_{2,1}^i$...	$fr_{2,10}^i$
---------	---------	--------------	-----	---------------	--------------	-----	---------------

I^j

f_1^j	f_2^j	$fr_{1,1}^j$...	$fr_{1,3}^j$...	$fr_{1,10}^j$	$fr_{2,1}^j$...	$fr_{2,10}^j$
---------	---------	--------------	-----	--------------	-----	---------------	--------------	-----	---------------

Luego de la aplicación del operador:

I^i

f_1^i	f_2^i	$fr_{1,1}^i$...	$fr_{1,10}^i$	$fr_{2,1}^i$...	$fr_{2,10}^i$
---------	---------	--------------	-----	---------------	--------------	-----	---------------

I^j

f_1^j	f_2^j	$fr_{1,1}^j$...	f_1^i	...	$fr_{1,10}^j$	$fr_{2,1}^j$...	$fr_{2,10}^j$
---------	---------	--------------	-----	---------	-----	---------------	--------------	-----	---------------

Operadores especializados para modelar macromutaciones

Recrudescencia: El operador de recrudescencia incrementa las probabilidades de recombinación, mutación y transposición de una porción de la población. Este operador se aplica en cada generación con una probabilidad p_{recru} y produce una reorganización radical fenotípica de los individuos a los cuales se les aplica. Los individuos para recrudescencia se seleccionan aleatoriamente con una probabilidad uniforme.

Crisis: Actúa igual que el operador de recrudescencia, pero se aplica a todos los individuos de la población, no sólo a un grupo, cada cierto número de generaciones en las etapas iniciales de la evolución.

3 FUNCION OBJETIVO

En el presente trabajo, para testear la validez del modelo se utilizó el Generador de Problemas de Test desarrollado por Morrison y De Jong [6]. Tal generador está concebido esencialmente para proveer un conjunto de funciones dinámicas, pero fijando algunos de sus parámetros se puede obtener un conjunto de funciones estáticas que permite manejar cualquier número de dimensiones, la función de fitness = función objetivo se define como:

$$f(x_1, \dots, x_n) = \max_{i=1, \dots, N} [H_i - R_i * \sqrt{(x_1 - x_{1i})^2 + (x_2 - x_{2i})^2 + \dots + (x_n - x_{ni})^2}]$$

donde (x_1, \dots, x_n) son los fenos activos de un individuo.

N especifica el número de conos en el ambiente.

(x_{1i}, \dots, x_{ni}) son las coordenadas del cono i , $x_{ki} \in [-1, 1]$ con $k = 1, \dots, n$.

H_i es la altura del cono i , con $H_i \in [H_{base}, H_{base} + H_{rango}]$

R_i es la pendiente del cono i , con $R_i \in [R_{base}, R_{base} + R_{rango}]$

Cada vez que el generador es invocado produce una morfología de “campos de conos” generada aleatoriamente en la cual los valores aleatorios para cada cono se calculan en un rango definido por el usuario.

Para generar un rango amplio de problemas estáticos de complejidad variable basta con especificar los parámetros N , H_{rango} , H_{base} , R_{rango} y R_{base} , cantidad de picos y dimensiones.

Un ejemplo del posible landscape generado aleatoriamente puede verse en la figura siguiente

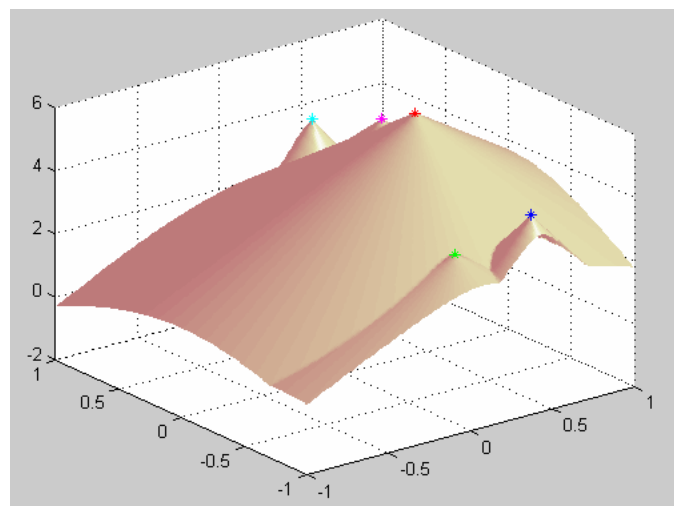


Fig. 1: Un posible campo de conos

4 DESCRIPCION DE LOS EXPERIMENTOS

Para realizar los experimentos se eligieron funciones de fitness con distintas características, en este trabajo sólo se presentan las siguientes:

Función	Dimensiones	Cantidad conos	Cono máximo	Valor óptimo
f1	5	5	4	5.714134
f2	5	30	2	5.755731
f3	10	5	3	5.058839
f4	10	30	30	5.977472

Para el caso de funciones de dos variables, que no se muestra, en el 100% de los experimentos se encontró el óptimo global, con errores despreciables.

En todos los experimentos se fijaron los siguientes parámetros para el algoritmo evolutivo:

Tamaño de la población: 100 individuos.

Inicialización de la población: Los fenos activos de cada individuo de la población se inicializan con las coordenadas del peor pico y los fenos redundantes en forma aleatoria (con el objetivo de estudiar si la población es capaz de escapar de un óptimo local).

Medida de performance: Como una medida del grado de evolución se considera el error entre el valor máximo encontrado (mejor individuo) y el valor óptimo global. La función de error empleada es:

$$\text{error} = (\text{valor óptimo} - \text{valor máximo encontrado}) / \text{valor óptimo} * 100$$

Cada uno de los experimentos se repitió 10 veces, informándose los valores medios de las 10 corridas. Todas las simulaciones ejecutaron hasta alcanzar las 10.000 generaciones y el objetivo de todos los experimentos fue determinar cuál es la performance de la representación elegida y de los nuevos operadores definidos sobre ella además de verificar si éstos son escalables, es decir, si a medida que se aumenta la cantidad de dimensiones y la cantidad de picos el modelo muestra un comportamiento semejante.

Varias corridas previas del algoritmo se realizaron con la finalidad de ajustar las probabilidades de los distintos operadores. La idea básica fue ir probándolos de manera incremental, esto es encontrar las mejores probabilidades para los operadores básicos de recombinación y mutación, luego incorporar el operador de recrudescencia y manteniendo las mejores probabilidades de recombinación y mutación halladas previamente variar las probabilidades del operador de recrudescencia y determinar cuál es la que mejor se comporta en el conjunto y, así siguiendo, hasta integrar todos los operadores. Con la mejor combinación de probabilidades así halladas se realizaron los experimentos para las funciones f1 a f4.

En todos los casos el operador de crisis se aplicó sólo durante la etapa inicial de la evolución, esto es durante las primeras 200 generaciones, probándose diferentes intervalos, es decir, cada 10 generaciones, posteriormente cada 20, cada 30, cada 40 hasta cada 50 generaciones.

En la sección siguiente se muestran y discuten los resultados preliminares obtenidos.

5 RESULTADOS

f1:

Probabilidades de:

Crossover 0.25, Mutación 0.2 ,

Transposición 0.01, Transición 0.01, Recrudescencia 0.3

Incrementos de probabilidades requeridos por el operador de Recrudescencia: Crossover 0.5, Mutación 0.5, Transposición 0.03

<i>Crisis</i>	Error Promedio Corridas	<u><i>Mejor Corrida</i></u>			<u><i>Peor Corrida</i></u>			Generación Promedio
		Error	Pico	ICO	Error	Pico	ICO	
10	0.041634	0.013873	4	70	0.130757	4	73	6642
20	0.062295	0.015418	4	66	0.137806	4	67	7617
30	0.046597	0.005838	4	62	0.097846	4	68	7088
40	0.025186	0.010598	4	72	0.053177	4	67	6020
50	0.043092	0.014825	4	67	0.139158	4	77	6554

donde ICO significa cantidad de individuos de la población ubicados en la cercanía del pico con el valor óptimo.

Es importante señalar que al usar el conjunto completo de operadores no disminuye tanto el error sino que lo que se logra es disminuir el número de generación donde se encuentra el mejor valor. Para esta función las expectativas puestas en el modelo fueron satisfechas, ya que si bien la población no alcanza el valor óptimo tiene un error promedio del 2,5% en todas las corridas, con errores que van del 1% (en el mejor caso) al 5% (en el peor). Desde el punto de vista de la optimización, un algoritmo que puede escapar de un óptimo local y acercarse al óptimo global con un error promedio del 2,5% y con el 70% de la población cercana a él permite decir que encuentra muy buenas soluciones.

f2:

Probabilidades de:

Crossover 0.25, Mutación 0.2 ,

Transposición 0.01, Transición 0.01, Recrudescencia 0.3

Incrementos de probabilidades requeridos por el operador de Recrudescencia: Crossover 0.5, Mutación 0.5, Transposición 0.03

<i>Crisis</i>	Error Promedio Corrida	<u><i>Mejor Corrida</i></u>			<u><i>Peor Corrida</i></u>			Generación Promedio
		Error	Pico	ICO	Error	Pico	ICO	
10	0.170521	0.099974	2	49	0.382487	2	43	8284
20	0.137804	0.063518	2	34	0.304026	2	37	7508
30	0.127577	0.058266	2	57	0.228308	2	42	8169
40	0.147120	0.068196	2	48	0.264870	2	47	8123
50	0.169265	0.076492	2	35	0.344950	2	41	8573

En este caso, el fitness landscape es bastante complejo puesto que si bien se mantiene el número de dimensiones, la cantidad de picos pasa de 5 a 30. Sin embargo, escalando el número de conos aun el algoritmo tiene un resultado satisfactorio puesto que en el mejor experimento, de los cinco que muestra la tabla, el error promedio es de aproximadamente el 12%, mientras que en el mejor caso es del 5% y en el peor es de 22% y casi el 60% de la población se ha movido del óptimo local (peor pico) hacia el pico donde se encuentra el valor del óptimo.

También puede notarse que para esta función el operador de crisis encuentra el mejor resultado cuando se lo aplica más seguido en las fases iniciales de la evolución y esto es de esperar si se tiene en cuenta que toda la población es la misma y es absolutamente necesario aumentar la diversidad genética en un landscape más accidentado. También como era de esperar el mejor valor es hallado en un mayor número de generaciones.

f3:

Probabilidades de:

Crossover 0.25, Mutación 0.2 ,

Transposición 0.01, Transición 0.01, Recrudescencia 0.3

Incrementos de probabilidades requeridos por el operador de Recrudescencia: Crossover 0.5,

Mutación 0.5, Transposición 0.03

<i>Crisis</i>	Error Promedio Corrida	<u><i>Mejor Corrida</i></u>			<u><i>Peor Corrida</i></u>			Generación Promedio
		Error	Pico	ICO	Error	Pico	ICO	
10	0.368563	0.138624	3	68	0.745332	3	65	7995
20	0.269290	0.163250	3	56	0.406872	3	70	8699
30	0.279354	0.117293	3	49	0.506398	3	61	8809
40	0.272639	0.142096	3	61	0.511483	3	55	8070
50	0.277925	0.117604	3	58	0.529491	3	64	7820

f4:

Probabilidades de:

Crossover 0.25, Mutación 0.2 ,

Transposición 0.01, Transición 0.01, Recrudescencia 0.3

Incrementos de probabilidades requeridos por el operador de Recrudescencia: Crossover 0.5,

Mutación 0.5, Transposición 0.03

<i>Crisis</i>	Error Promedio Corrida	<u><i>Mejor Corrida</i></u>			<u><i>Peor Corrida</i></u>			Generación Promedio
		Error	Pico	ICO	Error	Pico	ICO	
10	0.276323	0.130616	30	60	0.573208	30	54	8140
20	0.290862	0.147087	30	45	0.443645	30	66	8432
30	0.370465	0.129940	30	58	0.707901	30	52	8794

En el caso de f3 y f4 puede observarse que el porcentaje del error promedio de las corridas se duplica, esto es, pasa de un 12% a un 27% pero lo que aun resulta interesante es que alrededor del 55% de la población se encuentra en las cercanías del pico óptimo.

Esto permite, en primera instancia, aseverar que el modelo propuesto cumple con el objetivo deseado tal es sacar a la población de un óptimo local moviéndola hacia el global.

Una primera conclusión intuitiva, que se seguirá explorando como línea de investigación, con respecto al porcentaje de error es que éste podría disminuirse si se trabaja también sobre la autodaptación de los parámetros, tales como las probabilidades de aplicación de los operadores.

6. CONCLUSIONES

Una amplia variedad de problemas de optimización usando algoritmos evolutivos involucran la existencia de muchos óptimos locales. La posibilidad de salir de un óptimo local es una de las características deseables de los algoritmos de optimización.

El objetivo del trabajo está relacionado con la evolución de una población inicializada en el peor de los óptimos locales (en landscapes multimodales) y la posibilidad de alcanzar o acercarse, con un error aceptable, al óptimo global. A tal efecto se propuso una representación que incluye la existencia de fenos redundantes y la definición de un operador de recombinación acorde con la misma. Se utilizaron además operadores que implementan la macromutación indispensable para obtener, a partir de una población homogénea la necesaria diversidad genética como para que los individuos de la población exploren adecuadamente el espacio de soluciones. Tales operadores efectúan un aporte interesante y este primer intento es promisorio también en lo referente a la escalabilidad. Aun resta efectuar más experimentos aumentando la dimensionalidad de las funciones como también la cantidad de conos en la superficie.

Estos primeros resultados muestran que cuando la dimensionalidad y la complejidad del fitness landscape aumenta el algoritmo tiende a degradarse aumentando el porcentaje de error pero una parte importante de la población, alrededor del 60%, logra escaparse del óptimo local moviéndose hacia el pico donde se encuentra el valor global. Una primera intuición hace presuponer que el porcentaje de error con respecto al valor óptimo puede controlarse si se incluye un proceso de autoadaptación de parámetros (probabilidades de aplicación de los operadores) permitiendo que evolucionen junto con la población de soluciones. En esta línea se continuará la investigación.

Por otro lado, surge claramente que si el modelo permite escaparse de un óptimo local, el mismo podría adaptarse sin mayor esfuerzo para optimización en ambientes no estacionarios donde el fitness landscape varíe dinámicamente, implicando esto que el valor óptimo cambia en el tiempo. Ya se ha avanzado en esta línea de investigación y los resultados obtenidos son objeto de otra presentación simultánea con el presente artículo.

7. REFERENCIAS BIBLIOGRAFICAS

- [1] Gould S. J., "Ever Science Darwin, The Panda's Thumb, Hen's Teeth and Horses Toes, The Flamingo's Smile", PIW, Varsovia, Poland 1991.
- [2] Kwasnicka H., "Pleiotropy and Polygene Effects in Evolutionary Algorithms", V Proceedings of the Workshop on Intelligent Systems, Zakopane, Polish Academy of Science, Poland, June, 1997.

- [3] Kwasnicka H., "Redundancy of Genotypes as the Way for some Advanced Operators in Evolutionary Algorithms", VIVEK A Quartely in Artificial Intelligence, Vol. 10, N° 3, pp. 2 – 11, National Centre for Software Technology, Mumbai, India, July 1997.
- [4] Kwasnicka H., "Efficiency of Genetic Algoritms with Pleiotropy and Polygene Effects – Simulation Study, VI Proceedings of the Workshop on Intelligente Systems, Zakopane, Polish Academy of Science, Poland, June, 1998.
- [5] Mayr E., "Speciation and Macroevolution", Evolution 36 (6), 1982.
- [6] Morrison R. W., De Jong K.A., "A Test Problem Generator for Non-Stationary Environments", Proceedings of Congress on Evolutionary Computation, CEC99, V III, pp. 2047 – 2053, Washington DC, USA, IEEE Service Center, Piscataway, New Jersey, 1999
- [7] Szarsky H., "Mechanims of Evolution", PWN, Varsovia, Poland, 1986.